

Unity3D를 이용한 빗방울 렌더링 구현

이면재*, 김경남**

백석대학교 정보통신학부*, 성결대학교 교양교직부**

Implementation of Raindrop Rendering Using Unity3D Engine

MyounJae Lee*, Kyoung-nam Kim**

Division of Information & Communication, BaekSeok University*

Division of Liberal Art and Teaching, Sungkyul University**

요약 본 연구는 빗방울 렌더링에 관한 연구이다. 기존 게임에서 빗방울을 렌더링하는 경우에는 스프라이트 이미지를 이용하거나 대략적으로 빗방울이 맺힌 텍스처를 이용한다. 이 방법은 모든 빗방울이 비슷한 모양과 크기로 렌더링되기 때문에 플레이어들에게 실제감을 제공하기에는 한계가 있다. 이 한계를 개선하기 위해 본 논문에서는 빗방울이 물체와 접촉했을 때 표면 장력을 고려한 빗방울 생성 방법을 제안하고 이를 Unity3D 엔진을 이용하여 구현한다. 본 논문의 유용성을 보여주기 위해, 표면 장력 수식의 요소인 면적과 당기는 힘의 변화에 따른 빗방울 생성을 보여준다. 본 논문은 빗방울을 게임에서 렌더링 하는 경우 실제감 있는 게임 제작에 도움을 줄 수 있다.

주제어 : 빗방울 렌더링, 자연 현상 렌더링, 밀크 드롭, 표면 장력, 유니티3D

Abstract This research is the study of raindrop rendering. In case of rendering for raindrop in existing games, it is used on sprites images or roughly raindrops images using texture rendering. These methods are similar to the shape and size of all rendered raindrop. That's why players are limitations to provide a sense of reality. To overcome this limitation, this paper proposes a method for generating raindrop considering surface tension and contac angle, the amount of water, implements the raindrop using Unity3D engine. To demonstrate the usefulness of this paper, this paper shows the generated raindrop in accordance with the change in the area and pulling force in surface tension formula. This paper can help to provide the actuality in game in case of rendering the raindrop.

Key Words : Rain Rendering, Rendering of natural phenomena, Milk-Drop, Surface Tension, Unity3D

1. 서론

국내 게임 시장은 2011년 8조 8,047억, 2012년도에는 9조 7,525억으로 기록되었다. 그리고, 2013년에는 10조원을 돌파하고, 2015년에는 12조원에 근접하는 시장을 형성할 것으로 예상되고 있다[1].

2012년 국내 게임 시장 규모중 온라인 게임은 6조 7,839억 원의 매출을 달성하면서 전체 게임 시장의 69.6%를 점유했고, 모바일 게임은 2012년 8,009억원의 매출을 달성해 전체 게임 시장의 8.2%를 차지했다. 온라인 게임과 모바일 게임이 국내 게임 산업의 성장의 중요

Received 17 December 2013, Revised 17 January 2014

Accepted 20 January 2014

Corresponding Author: Kyoung-nam Kim(Division of Liberal Art and Teaching, Sungkyul University)

Email: hsfruitr@sungkyul.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

축을 형성하고 있는 것이다.

이러한 국내 게임 산업의 성장에는 게임 스토리, 게임 커뮤니티, IT 인프라 등 여러 가지 요인들이 플레이어들을 게임 세계로 유인했기 때문이다. 이러한 많은 요인들 중에서 게임속 세상이 실제 세상과 유사한 환경을 제공하고 있는 것도 플레이어의 몰입을 돕는 큰 원인이 될 수 있다. 즉 실제의 시각적인 요소들과 청각적인 요소들을 게임내에서 표현하기 위한 기술의 발달이 뒷받침되고 있었기 때문이다.

플레이어들의 청각 요소와 시각 요소를 만족시키기 위해서는 컴퓨터 그래픽스 기술과 사운드 기술을 처리할 수 있는 하드웨어 기술의 발달과 연구가 뒷받침 되어야 한다.

특히 청각적인 요소들보다 시각적인 요소들을 중심으로 한 렌더링 연구들이 과거부터 많이 진행되어져 왔다. 게임은 실시간에 플레이되기 때문에 다음 진행에 방해되지 않을 범위에서 사실적인 렌더링 처리를 해야 한다. 즉 실시간 수행이라는 게임 진행의 목표를 수행하기 위해 어느 정도 사실적인 결과물을 포기해야한다[2].

이러한 연구에는 화염에 대한 비사실적인 렌더링 방법[3], 카툰 스타일을 표현하기 위하여 3차원 물체의 음영을 단순화 시키는 카툰 렌더링 방법을 이용하여 유체를 구현한 연구[4], 바다 렌더링[5], 식물 표면 렌더링[6], 번개[7], 파티클 입자로 일종의 방사체를 이용하여 난수를 발생시켜서 구름을 생성하는 연구[8] 등의 다양한 자연 소재들을 대상으로 한 연구 등이 있다. 또한, 현재 많은 게임들에서는 실제세계의 계절과 기후에 관계된 것을 실제적으로 표현하기 위해 많은 방법들이 연구되어지고 있다. 비, 눈, 바람 등과 봄, 여름, 가을, 겨울 같은 것이 이에 속한다. 본 논문은 빗방울 렌더링에 관한 연구이다.

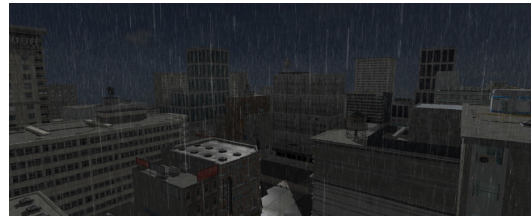
현재 많은 게임들에서는 빗방울이 하늘에서 떨어지는 경우와 물체에 닿았을 때 퍼지는 경우를 주로 표현하고 있다. 특히 빗방울이 물체에 닿았을 때 대부분의 게임에서는 미리 물방울이 맺혀진 표면을 만들어 놓고 이를 렌더링하는 방법을 사용한다[9]. 그러나, 현실세계에서 빗방울은 충돌하는 재질과 빗방울의 크기, 그리고 표면 장력(Surface Tension)에 따라 다른 모양을 갖는다[10]. 표면 장력은 액체의 자유 표면에서 표면을 작게 하려고 작용하는 액체 내부의 힘을 말한다[14]. 이 힘이 클수록 빗방울은 구형을 이룬다.

빗방울의 모양은 충돌하는 재질과 빗방울의 크기, 그리고 표면 장력에 달라질 수 있는데, 이 중에서 본 논문은 표면 장력을 이용한 빗방울 생성 방법을 구현하기 위한 것이다. 구현 엔진은 Unity3D 엔진 Pro 4.2를 이용한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 설명하고 3장과 4장에서는 각각 제안 방법과 구현 결과를 기술한다. 5장에서는 결론 및 추후 연구 방향을 논한다.

2. 관련 연구

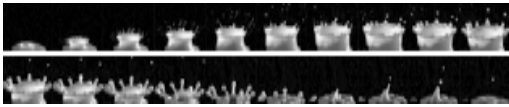
현재 게임에서 비와 빗방울을 처리하는 경우에는 하늘에서 빗방울이 떨어지는 모습을 보여주거나 미리 물체의 표면에 빗방울이 닿아 모양이 변화되는 텍스처를 미리 만들어 놓은 스프라이트 이미지를 연속적으로 재생함으로써 빗방울이 물체의 표면에 닿은 효과를 발생시킨다. [Fig. 1]은 게임에서 비가 내리는 화면을 보여준다[11].



[Fig. 1] Rain scene rendering

현재 게임에서 빗방울 렌더링 방법은 크게 밀크 드롭(Milk Drop) 방식과 표면에 빗방울이 맺혀있는 이미지 자체를 렌더링하는 방법, 이 2가지 방법으로 주로 표현된다.

밀크 드롭 방법에서는 사전에 렌더링이 이미 완성된 고품질들의 이미지들을 단계적으로 보여준다[2]. 이 방식은 표면과 빗방울이 충돌할 때 생기는 자연적인 현상을 나타내기 위해 일정한 패턴의 이미지를 미리 제작해 놓은 후 빗방울이 물체와 충돌할 때 마다 이 이미지들을 렌더링 하는 방식이다. 이러한 방법은 렌더링 처리가 간단하지만 세부적인 처리가 부족한 것이 단점이다. 이러한 몇몇 부족한 부분이 현실적인 게임을 추구하는 방향에선 지루한 느낌을 줄 수 있다. [Fig. 2][9]는 밀크 드롭 방식([Fig. 2] (a))과 게임에서 처리된 예([Fig.2] (b))를 보여준다.



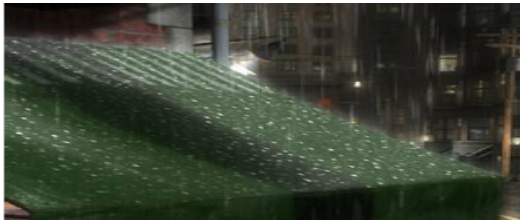
(a) Milk-Drop Rendering Method



(b) An Example of Milk Drop Rendering Method

[Fig. 2] Milk-Drop Rendering

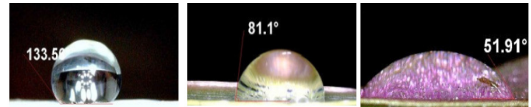
표면 렌더링 방법은 여러 빗방울이 표면에 맺혀 있는 것을 미리 이미지로 만들어 놓고 이를 렌더링하는 방법으로 렌더링 처리 시간이 적게 소요되지만 모든 물방울이 동일한 모양으로 처리되므로 출력되는 렌더링 이미지는 자연 세계에서 빗방울과는 다소 거리가 있는 결과가 나올 수 있다. [Fig. 3]은 빗방울이 표면에 부딪혔을 때 렌더링 된 게임 화면[9]를 보여준다.



[Fig. 3] Raindrop surface rendering

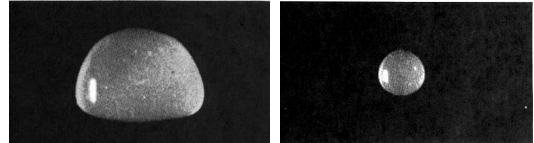
두 방법 모두 미리 정해진 이미지들을 반복하여 렌더링하는 패턴으로 빗방울을 처리하기 때문에 물체 표면에 빗방울이 부딪혔을 때 접촉각에 따른 빗방울의 모양, 빗방울의 양에 따른 빗방울의 모양, 표면 장력 등을 고려하고 있지 않아서, 이 방법들은 플레이어들에게 현실감을 제공하기에는 한계점을 갖고 있다.

현실 세계에서 빗방울은 빗방울이 충돌하는 재질과 빗방울의 크기에 따라 표면 장력이 다르기 때문에 생기는 빗방울의 모양이 다르다[10]. [Fig. 4][13]는 빗방울이 충돌하는 재질이 다름에 따라 달라지는 물방울의 모양의 차이를 보여준다.



[Fig. 4] The contact angle changes according to the type of plant

[Fig. 5][14]는 빗방울의 양에 따른 모양 변화를 보여준다.



(a) A small amount of raindrop (b) A large amount of raindrop

[Fig. 5] The shape of raindrops changes According to the amount of raindrop

3. 제안 방법

본 장에서는 표면 장력을 고려한 빗방울 모양을 생성하기 위해 표면 장력 수식을 설명한다.

식 (1)은 표면 장력을 나타내는 공식[9]이다. γ 는 표면 장력을 의미하고 J 는 물체 고유의 잡아당기는 힘이다. m 은 면적을 나타낸다. 이 식은 임의의 재질과 접촉했을 때 빗방울 모양은 접촉 부위 면적이 적고 표면 장력을 만드는데 요구되는 에너지가 클수록 표면 장력이 커져서 구체 모양에 가깝게 됨을 나타낸다. 같은 표면 장력을 가진 상태로 접촉 면적만 크게 했을 경우 표면 장력은 작게 되어 기존 모양보다 넓게 퍼지는 모양이 된다는 것을 알 수 있다.

$$\gamma = J/m^2 \quad \dots \quad (1)$$

빗방울(<Table 1>의 Water)은 접촉각의 각도와 표면 장력에 따라 다른 모양이 생성된다. <Table 1>은 실온에서의 빗방울의 표면 장력과 접촉각을 보여준다.

<Table 1> [5] Water Surface Tension and Contact Angle Measured Contact Angles of Water at 23 °C

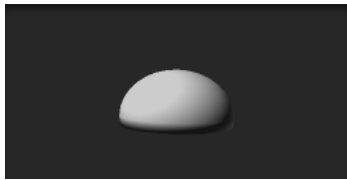
| Compound | Formula | Surface Tension (dyne/cm) ^b | Contact Angle (°C) |
|----------|------------------|--|--------------------|
| Water | H ₂ O | 71.99 | 31.6 |

4. 실험 결과

구현한 PC의 메모리는 5.00GB 이며, CPU는 Intel Core(TM) i5-2430M이며, 운영 체제는 Window 7이고, 구동한 게임 엔진은 Unity3D로 4.2 버전이다. Unity3D 엔진을 사용한 이유는 제작이 용이하고 엔진 구입 비용이 저렴하고 이식성이 높은 장점 때문에 국내 게임 개발자들이 가장 선호하는 엔진이기 때문이다[15].

본 실험에서 사용한 기본적인 빗방울 모양은 자연의 상온 속 빗방울의 접촉 모양을 이용한다. [Fig. 6]은 이를 보여준다. 이 모양은 [Fig. 4]와 [Fig. 5]를 기본으로 하여 최대한 유사하게 만든 것이다. 이 모양을 기본으로 표면 장력 식의 각 항인 빗방울의 면적과 잡아당기는 힘 값에 변화를 주어 생성된 빗방울을 비교한다.

여러 개의 빗방울을 생성하는 경우 [Fig. 6]을 기본 모양으로 생성한다. 이와 같이 1개의 소스 물체를 가지고 여러 개의 물체를 생성할 수 있도록 하는 것을 프리팹(prefab)이라고 한다. [Fig. 6]을 빗방울 프리팹이라고 하며 이 프리팹 안에는 이 빗방울의 움직임과 크기, 그리고 잡아당기는 힘 등을 조절할 수 있는 스크립트가 포함되어 있다. 스크립트는 각 프레임에서 객체가 해야 할 일을 기술한다. 객체 지향 프로그래밍에서 객체가 멤버 데이터와 함수가 있는 것과 유사하다.



[Fig. 6] default prefab

<Table 2>는 구현 스크립트의 초기 설정을 보여준다. 이 스크립트는 [Fig. 6]의 빗방울 프리팹에 포함되어 있으며 이 프리팹을 이용하여 빗방울을 생성할 때 초기값을 갖고 있다. Unity3D에서 Start() 함수는 객체 초기화와 관련된 함수이다.

라인 (1)부터 (3)까지는 크기 변화에 따른 물방울의 모양 변화를 비교하기 위하여 빗방울의 충돌 면적 값으로 2부터 4까지의 값을 할당하는 과정이다. 라인 (4)에서는 표면 장력 수식의 J값을 설정한다. 이 값은 [Fig. 6]의 디

폴트 프리팹을 만들 때의 J값 즉 3으로 설정한다.

<Table 2> Start()

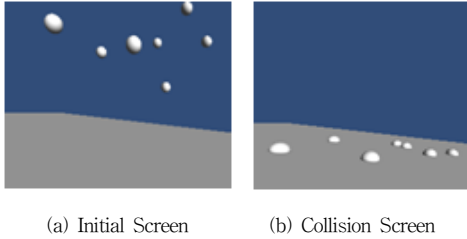
| |
|-------------------------|
| (1) default_value = 2f; |
| (2) Middle_value = 3f; |
| (3) Large_value = 4f; |
| (4) J_value = 3f; |

<Table 3>은 물체와 충돌하는 경우 빗방울을 생성하는 함수를 설명한다. 라인 (1)의 OnCollisionEnter() 함수는 빗방울이 충돌했을 때 호출되는 콜백(CallBack) 함수로 충돌된 재질을 식별하여 해당 재질에 맞는 빗방울 모양을 생성하는 함수이다. 라인 (3)에서는 표면 장력을 계산한다. J_value와 default_value는 <Table 2>에 정의되었다. 라인 (4)에서 라인 (2)의 표면 장력 값을 빗방울의 3차원 벡터 z값에 곱하여 개략적인 빗방울 모양을 생성한다. 라인 (6)은 물체와 충돌한 빗방울 모양을 다음 프레임에서 보여주기 위하여 물체와 충돌하기 전 빗방울 객체 즉, 현재 빗방울 객체를 제거하는 과정이다. 라인 (7)에서는 라인 (3)에서 (5)까지의 과정을 통해 계산된 값들을 이용하여 물체와 충돌한 빗방울을 생성시키는 과정이다. 이때 빗방울은 충돌된 위치에서 생성된다. Instantiate()함수는 프리팹에서 1개의 빗방울 객체를 생성하는 함수이다. 중간 크기의 빗방울인 경우 <Table 2>의 Middle_value에 저장된 크기로 빗방울을 생성시킨다(라인 (8)).

<Table 3> OnCollisionEnter()

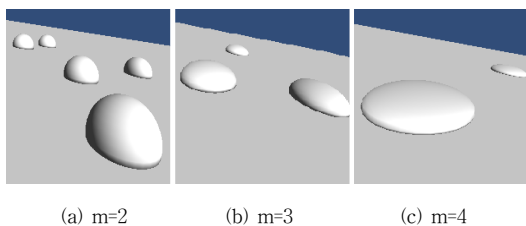
| |
|---|
| (1) function OnCollisionEnter(coll : Collision){ |
| (2) if(coll.gameObject.tag == "RainDrop_Small"){ |
| (3) ST = J_value/(default_value*default_value); |
| (4) prefab.localScale = Vector3(default_value, |
| (5) default_value, ST*default_value); |
| (6) Destroy(coll.gameObject); |
| (7) Instantiate(prefab, coll.gameObject.....); |
| (8) }else if(coll.gameObject.tag == "RainDrop_Medium"){ |
| (9) ST = J_VALUE/(Middle_value*Middle_value); |
| (10) prefab.localScale =Vector3(Middle_value, |
| (11) Middle_value, ST*Middle_value); |
| (12) |
| (13) }else if(coll.gameObject.tag == "RainDrop_Large"){ |
| (14) |

[Fig. 7]은 빗방울 면적이 2인 경우 즉 <Table 2>의 default_value를 이용하여 생성된 빗방울의 모양을 보여준다. [Fig. 7] (a)는 초기 화면이고 (b)는 재질과 충돌했을 경우 생성된 빗방울을 보여준다. 빗방울이 떨어진 위치와 동일한 자리에서 빗방울이 생성되는 것을 볼 수 있다. 카메라의 위치에 따라 빗방울의 원근감이 적용되어 있다.



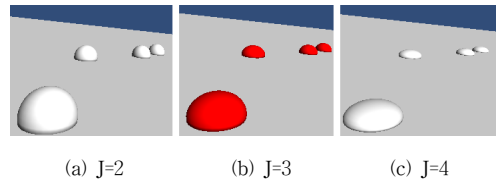
[Fig. 7] Experimental Result

[Fig. 8]은 재질과 충돌되는 빗방울 면적 값, 즉 m 에 따른 빗방울 모양 변화를 보여준다. 식 (1)에서도 살펴본 듯이 빗방울의 충돌 면적이 클수록 표면 장력이 약해져서 옆으로 퍼진다. 충돌 면적이 2인 경우 생성된 [Fig. 8] (a)의 빗방울은 충돌 면적이 3인 경우와 4인 경우 보다 구 모양을 형성한다. 이는 충돌 면적이 2인 경우가 충돌 면적이 3인 경우와 4인 경우보다 표면 장력이 크기 때문이다.



[Fig. 8] Comparison of the raindrop shape by the size(m)

빗방울의 충돌 면적뿐만 아니라 식 (1)의 J 값, 즉 물체 고유의 잡아당기는 힘에 의해서도 빗방울 모양은 변한다. [Fig. 9]는 J 값에 따른 빗방울 모양 비교이다. 이때 m 값은 2로 고정되었다. J 값이 커질수록 표면 장력이 커져서 빗방울과 만나는 접촉 부분이 구형 형태가 된다.



[Fig. 9] comparison of the raindrop shape by the pulling force(J)

5. 결론

기존 게임에서는 빗방울을 렌더링 하기 위하여 밀크드롭 방식이나 빗방울이 맺힌 텍스처를 렌더링하였다. 이 방법은 비교적 구현이 단순하지만 빗방울 각자 단위로 렌더링 되지 않기 때문에 플레이어들에게 실제감을 제공하기에는 한계가 있다.

본 연구에서는 비오는 날의 실제감을 플레이어에게 제공하기 위하여 표면 장력을 이용한 빗방울 렌더링 방법을 제안하고 이를 Unity3D 엔진을 이용하여 구현하였다. 구현 결과, 각 빗방울 단위로 렌더링 됨을 보여주었다. 표면 장력의 향, 즉 빗방울의 충돌 면적과 끌어 당기는 힘의 변화에 따라 빗방울 모양이 변함을 보여주었다.

제안 방법으로 비오는 날의 빗방울을 게임에서 렌더링 하는 경우 간단한 스크립트로 각 빗방울을 보다 현실적으로 표현할 수 있어서 플레이어들에게 현실감을 제공할 수 있을 것이라고 판단된다.

추후에는 접촉각과 온도, 그리고 빗방울의 양에 따라서 빗방울의 모양이 어떻게 변화되는지를 연구하고 이를 구현할 계획이다.

REFERENCES

- [1] KOCCA, Republic of Korea Game White Paper, 2013.
- [2] Kyuman Jeong, Seong-Whan Kim, "Application of non-photorealistic rendering techniques for computer games", Journal of Korea Computer Game Society, No.14, 2008.
- [3] Heekyung Yang, Kyungha Min, A Non-Photorealistic Rendering Technique of Flame for Games, KCGS,

Vol.25, No.4, 2012.

- [4] Kwang-tae Kim, Kyoung-ju Park, "Environment-Adaptive and Real-Time Rendering of Fluids for Cartoon Style Game", Journal of KGS, Vol.12, No.5, 2012.
- [5] Ho Min Lee, Won Hyung Lee, "A Frustum Based Real Time Ocean Rendering with GPU", Journal of KCGS, Vol.10, pp.43-48, 2007.
- [6] Young-Min Kang, "Realistic Rendering of Woven Surface using Procedural Bump Mapping", Journal of KCGS, Vol.10, No.3, pp.103-111, 2010.
- [7] SungBae Park, GyuHwan Oh, "A Study on Real-Time Lightning Simulation for Smart Device", Journal of KGS, Vol.13 No.4, pp.35-46, 2013.
- [8] Sung-Uk Lee, "Generating clouds System for the emotional online game", Journal of KCGS, Vol.26 No.1, 2013.
- [9] Natalya Tatarchuk, Artist-Directable Real-Time Rain Rendering in City Environments, Proceeding NPH'06 Proceedings of the Second Eurographics conference on Natural Phenomena, pp.61-73, 2006.
- [10] Zhiguang Guo, Weimin Liu, "Biomimic from the superhydrophobic plantleaves in nature: Binary structure and unitary structure", Plant Science 172, pp.1103-1112, 2007.
- [11] Carles Creus Lopez, Gustavo Ariel Patow, "Realistic Rain Rendering in realtime", Computer & Graphics, Vol.37, Issue 1-2, pp. 33-40, 2013.
- [12] Jasper, J. J. J. Phys. Chem. Ref. Data 1972, 1, 841-1009.
- [13] S.J Kim, "Study on the contact angle of water droplets formed on the leaf surface", The 54th National Science Exhibition, pp.18-25, 2008.7
- [14] Benny Lautrup, Physics of Continuous Matter, pp.69-95, CRC Press, 2011.
- [15] Myoun-Jae Lee, "A Study on Game Production Education through Recent Trend Analysis of 3D Game Engine", Journal of the Korea Convergence Society, Vol.4, No.1, pp.7-12, 2013.

이 먼 재(Lee, MyounJae)



- 1994년 2월 : 홍익대학교 전자계산학과(석사)
- 2006년 2월 : 홍익대학교 전자계산학과(박사)
- 2009년 3월 ~ 현재 : 백석대학교 정보통신학부 멀티미디어 전공 조교수
- 관심분야 : 기능성 게임, 게임 프로그래밍, 게임엔진
- E-Mail : davidlee@bu.ac.kr

김 경 남(Kyoung-nam Kim)



- 1994년 2월 : 홍익대학교 미술대학 회화과 졸업(학사)
- 1997년 2월 : 홍익대학교 대학원 회화과 졸업(석사)
- 2011년 8월 : 중앙대학교 첨단영상대학원 영상공학 졸업(박사)
- 2000 ~ 2006년 : 호서대학교 겸임교수
- 2008 ~ 2009년 : 한국콘텐츠진흥원 과장
- 현재 : 성결대학교 교양교직부(멀티미디어) 조교수
- 관심분야 : 디지털콘텐츠, 디지털아트, 게임
- E-Mail : hsfuit@sungkyul.ac.kr